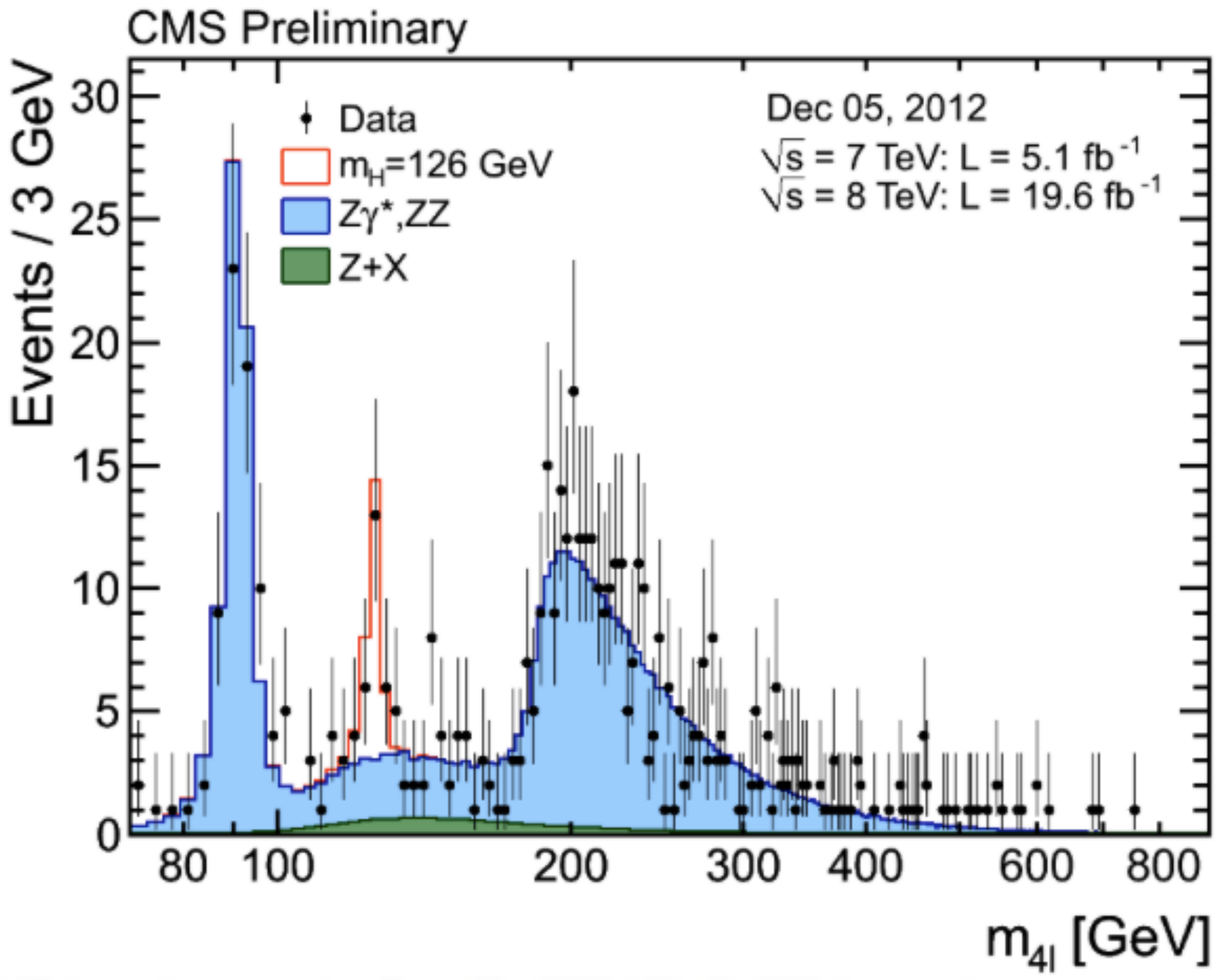
Infrastructure for Large Scale HEP data analysis

Matteo Cremonesi
FNAL

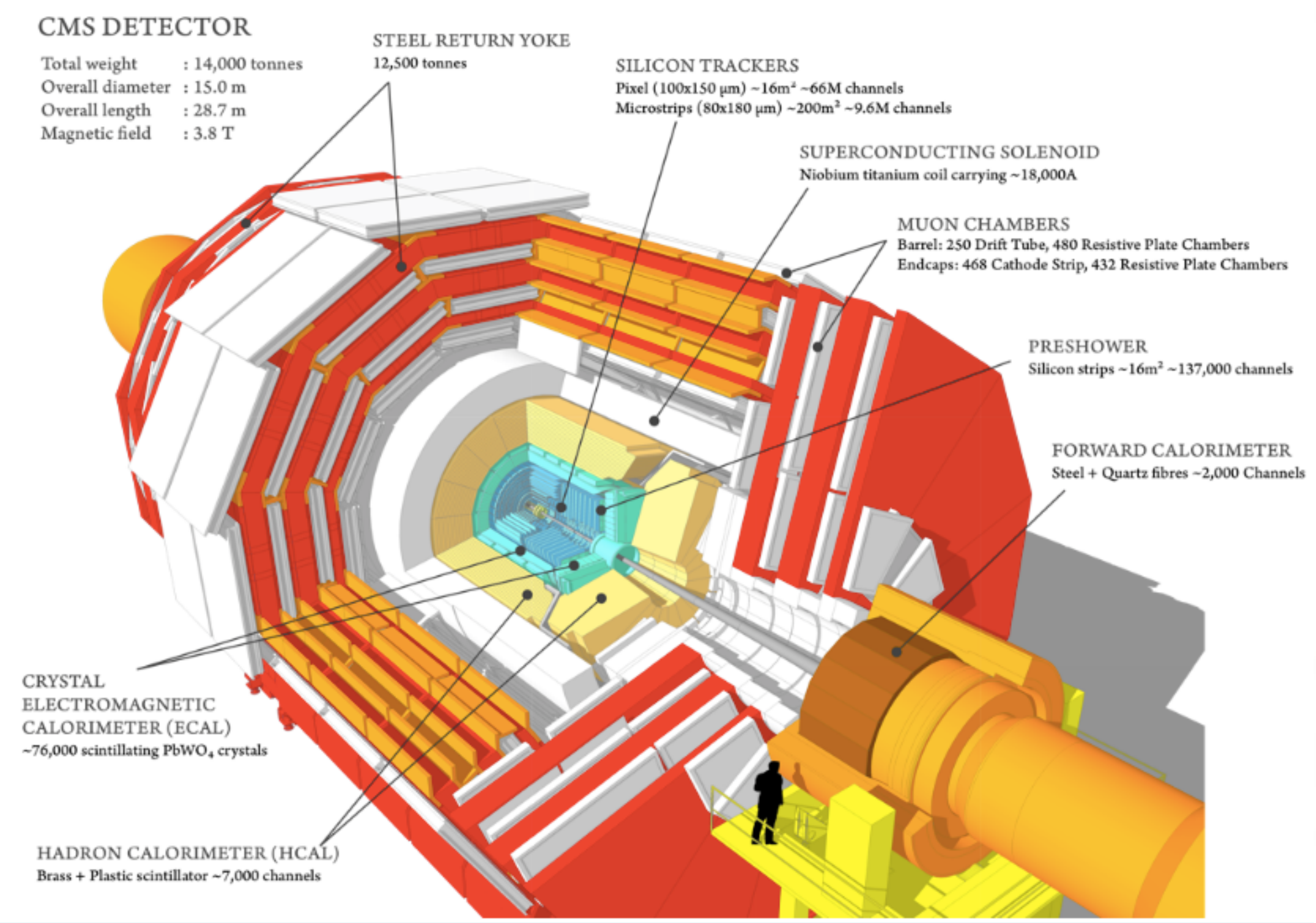
DS@HEP - May 11, 2017

Experimental Particle Physics

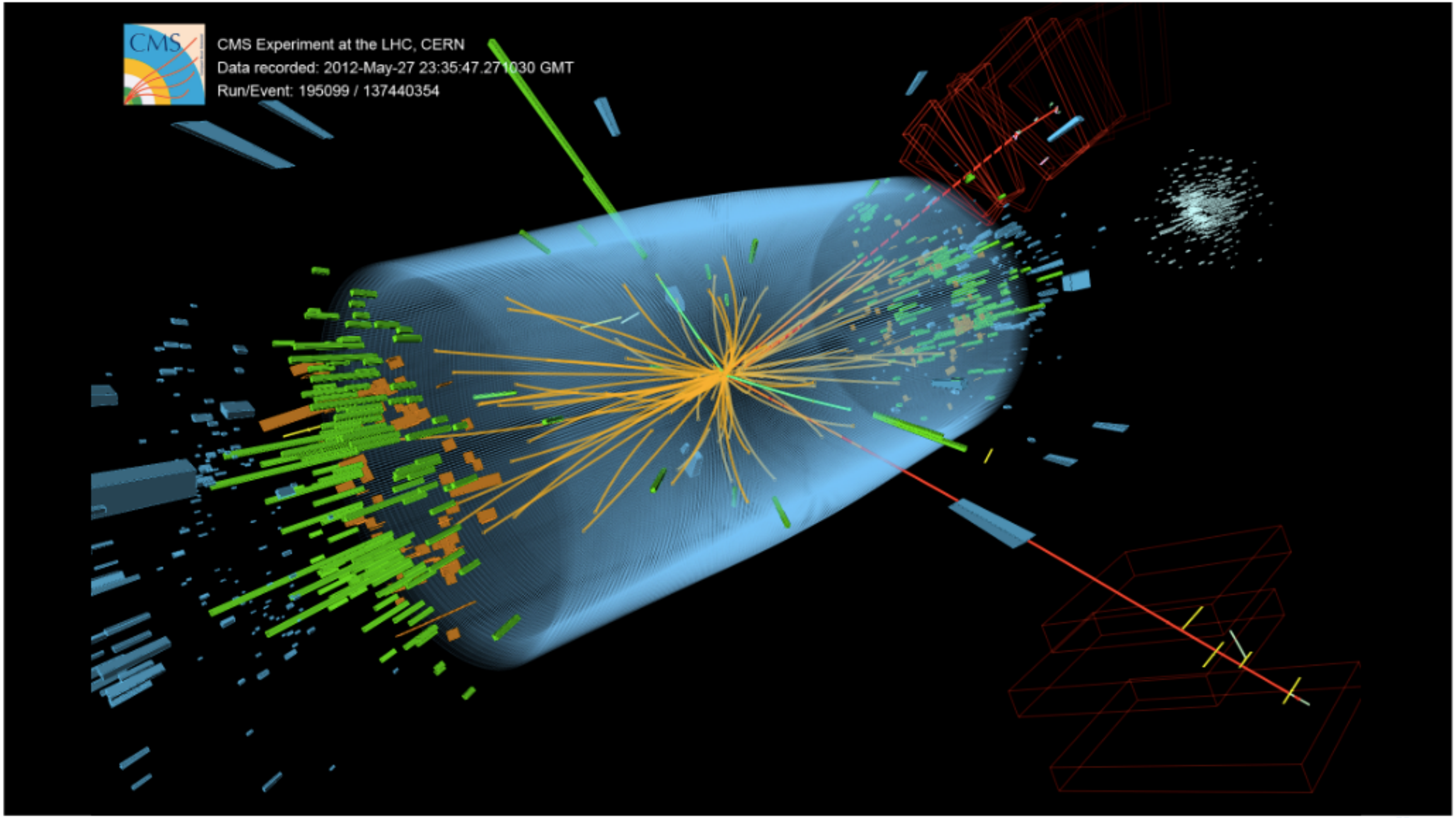


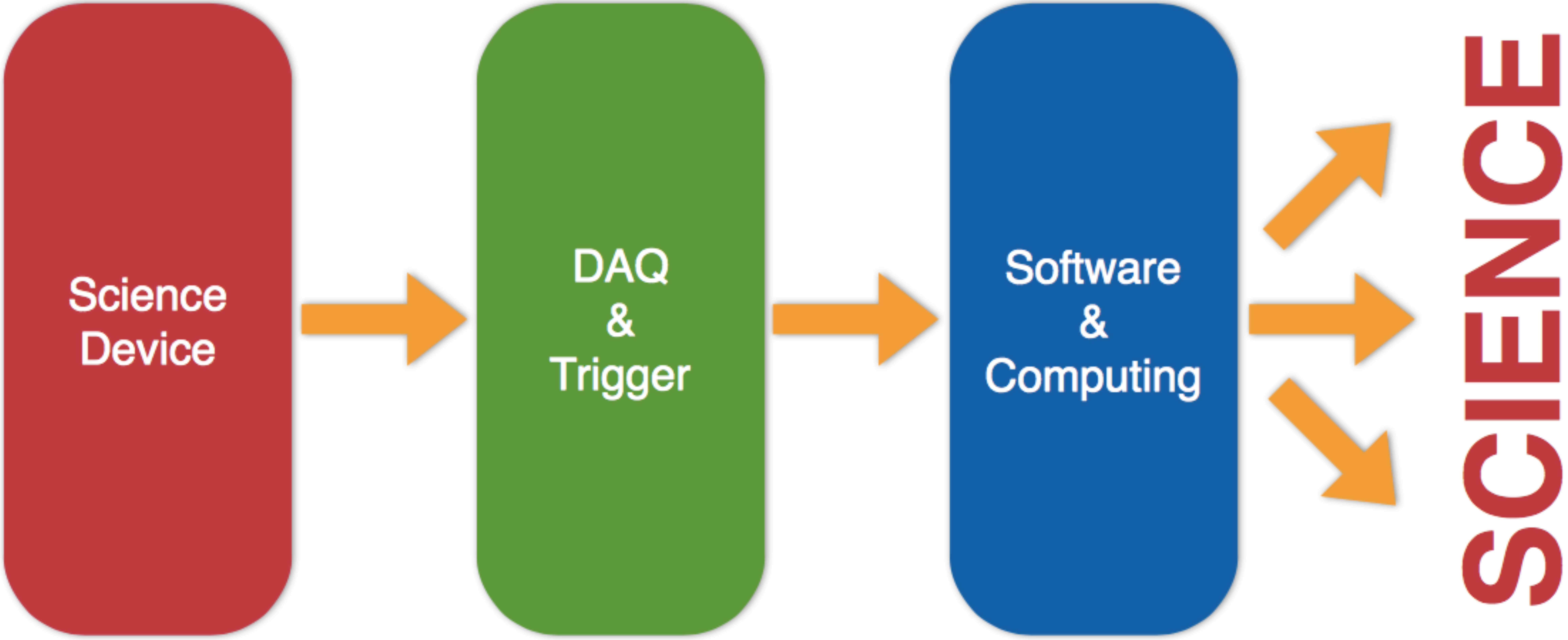
- Detect particle interactions (data), compare with theory predictions (simulation)
  - **Black dots: recorded data**
  - **Blue shape: simulation**
  - Red shape: simulation of new theory (in this case the Higgs)

Data Events

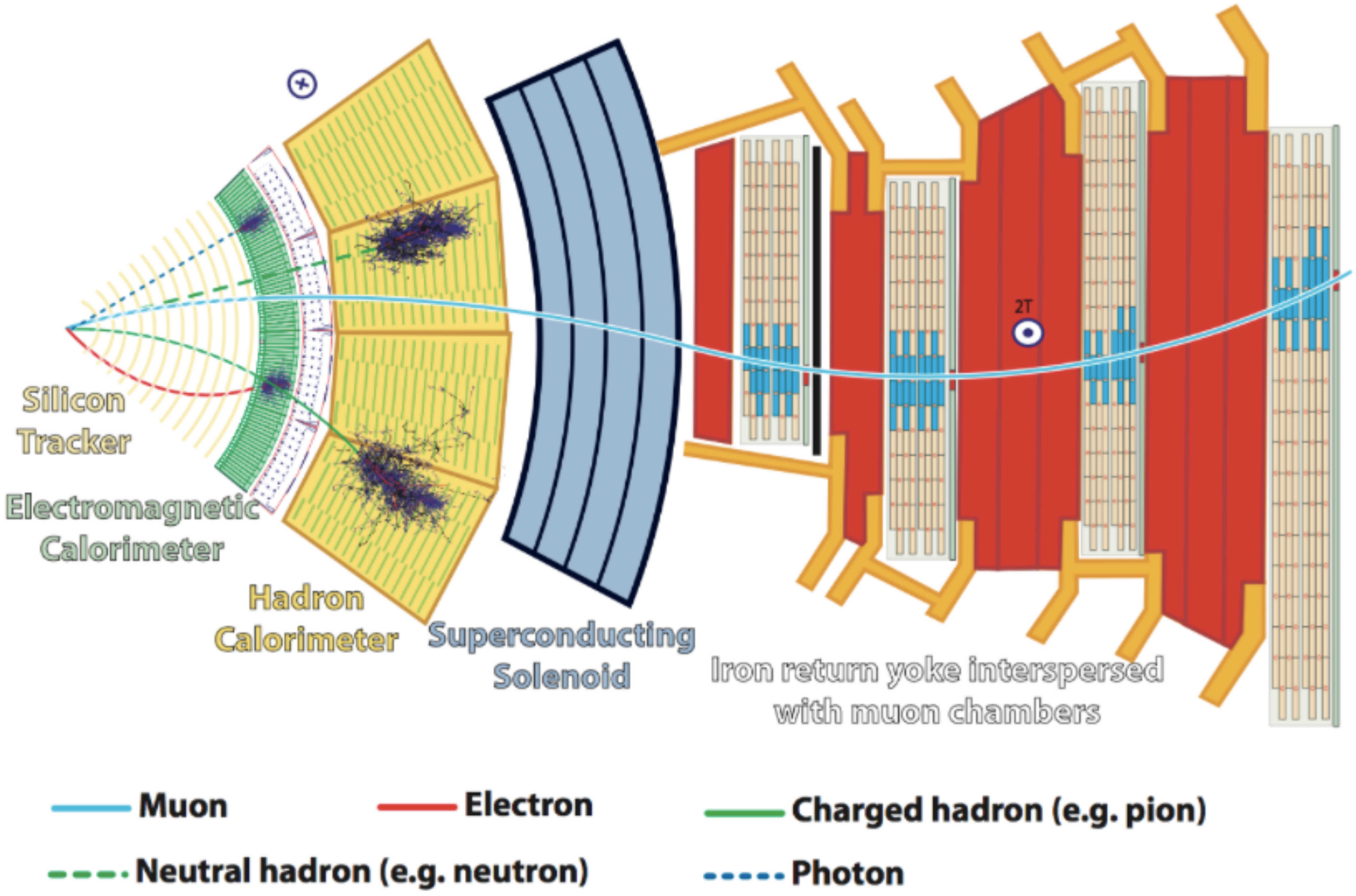


- **Particle detection** = record physics quantities (energy, flight path) of particles produced in a collision
  - Quantities **measured** from the interaction of particles and the different detector components
    - 100 Million individual measurements
  - All measurements of a collision together are called **event**


CMS
CMS Experiment at the LHC, CERN
Data recorded: 2012-May-27 23:35:47.271030 GMT
Run/Event: 195099 / 137440354


Science Device
DAQ & Trigger
Software & Computing
SCIENCE

Event Reconstruction

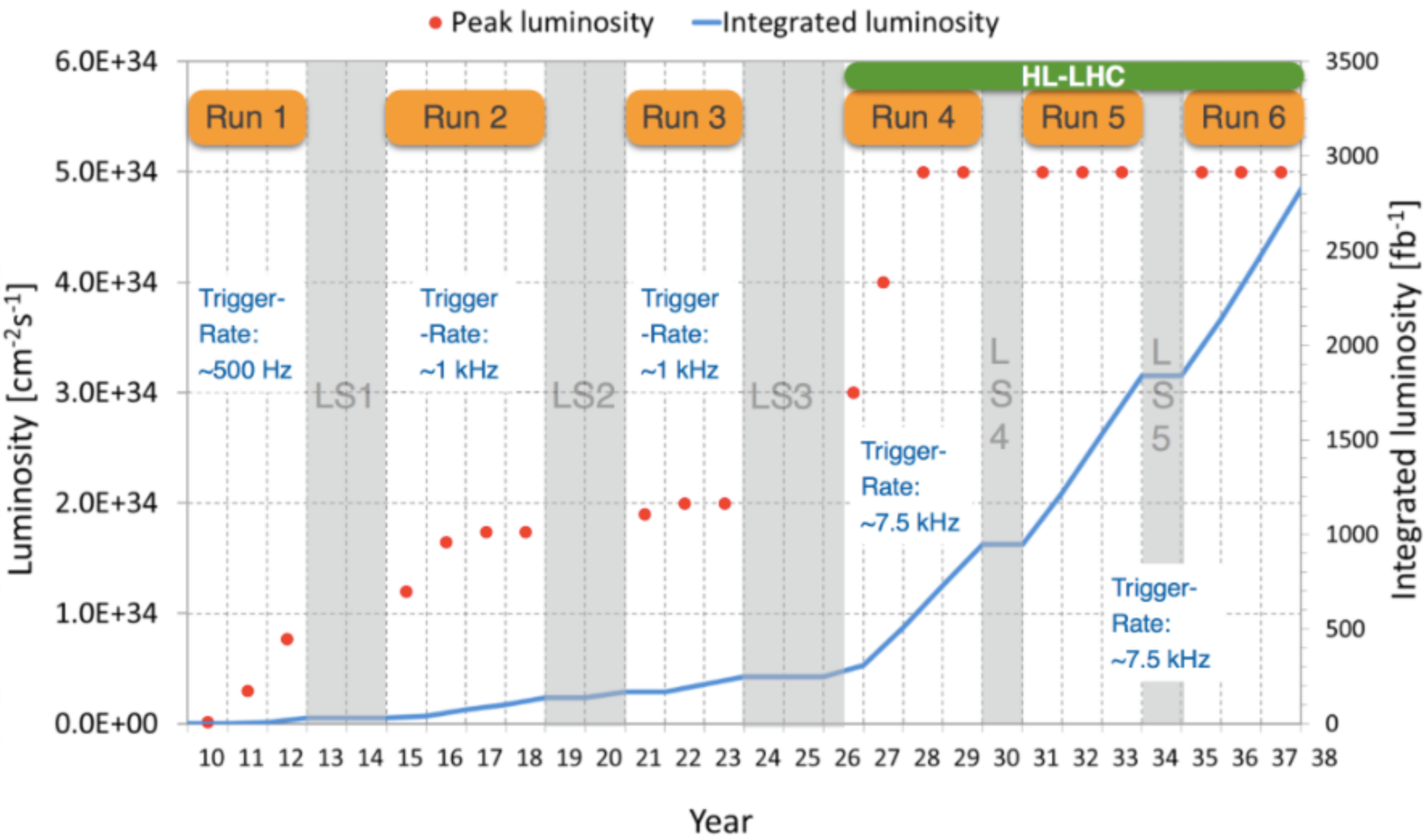


- Detector signals (and equivalent simulated signals) need to be reconstructed to learn about the particles that produced them

- The reconstructed events are then used for analysis


Device
Simulation
RAW Data
Algorithms to reconstruct data
RECO Data
Analysis software
PLOTS

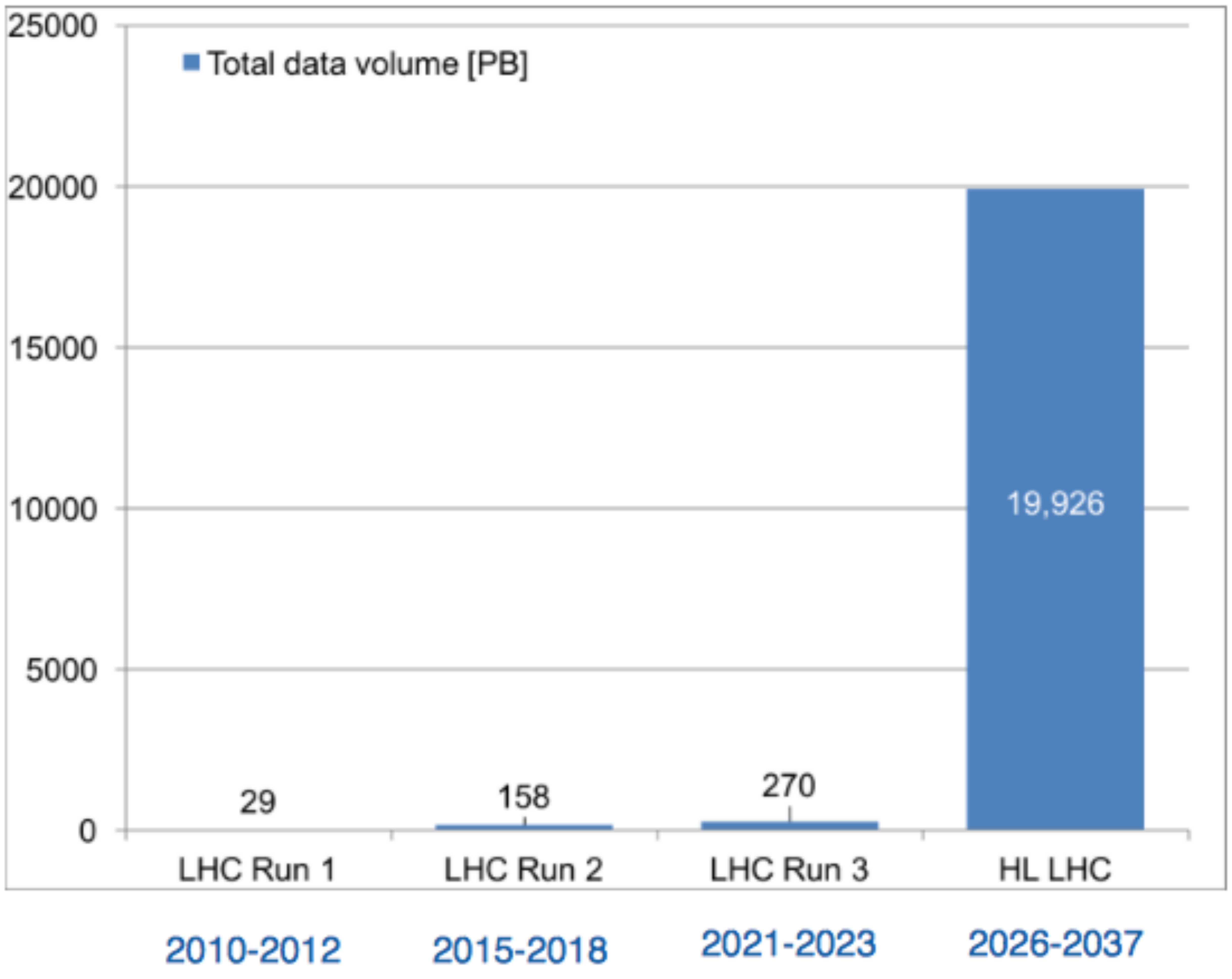

Device
Simulation
RAW Data
Algorithms to reconstruct data
RECO Data
Analysis software
PLOTS
Central
Chaotic



Peak luminosity
Integrated luminosity
HL-LHC
Run 1
Run 2
Run 3
Run 4
Run 5
Run 6
LS1
LS2
LS3
LS4
LS5
Trigger-Rate: ~500 Hz
Trigger-Rate: ~1 kHz
Trigger-Rate: ~1 kHz
Trigger-Rate: ~7.5 kHz
Trigger-Rate: ~7.5 kHz
Luminosity [cm-2s-1]
Integrated luminosity [fb-1]
6.0E+34
5.0E+34
4.0E+34
3.0E+34
2.0E+34
1.0E+34
0.0E+00
3500
3000
2500
2000
1500
1000
500
0
10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38
Year

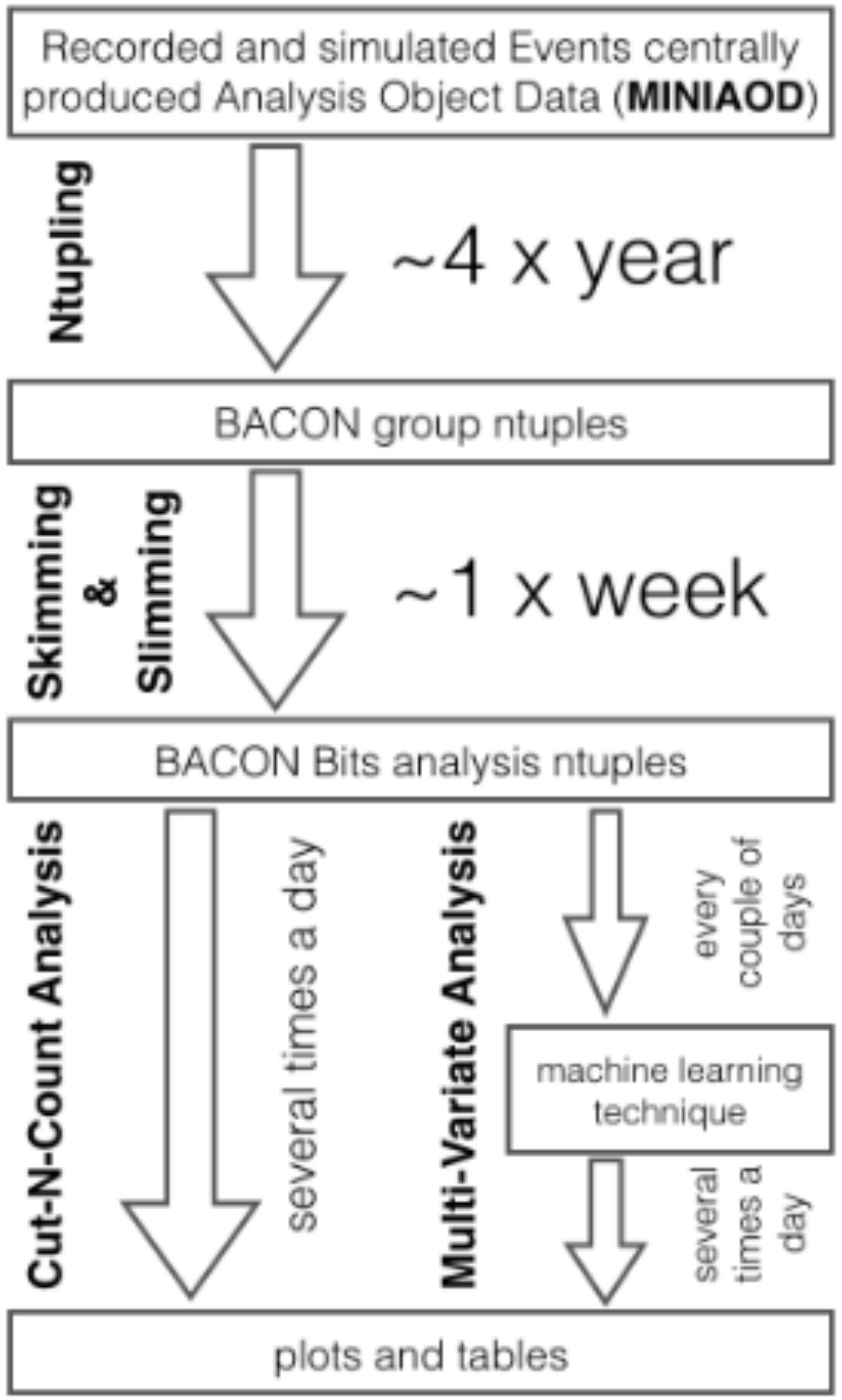
Data Volume @ HL-LHC

- Extract physics results will require to handle/analyze a lot more data
  - you cannot afford chaos anymore
- Explore industry technologies as suitable candidates for user analysis



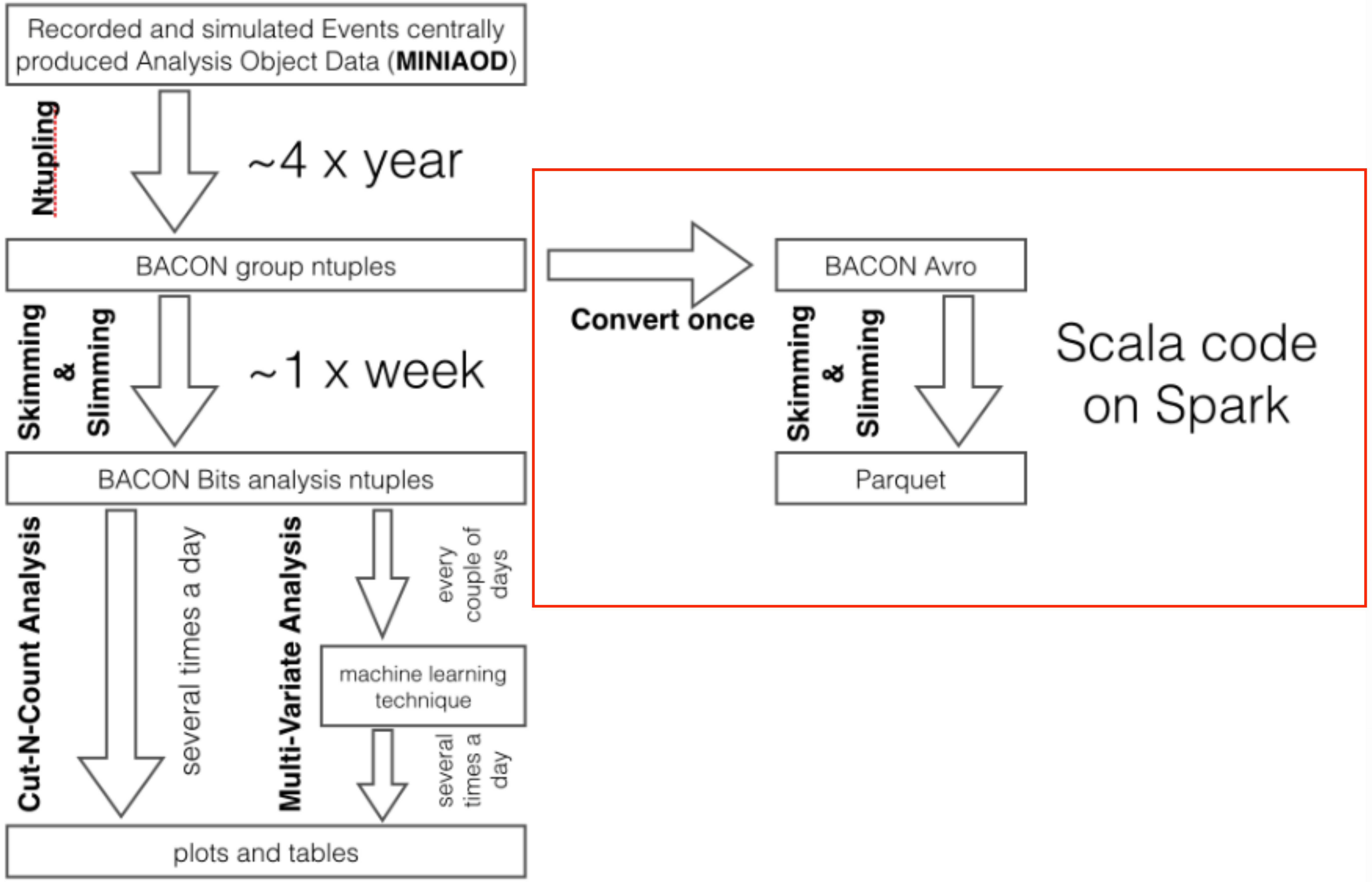
Current Analysis Workflow

- Input:
  - Centrally produced output of reconstruction software, reduced content optimized for analysis
  - Apply updated CMS reconstruction recipes
  - Too big for interactive analysis
- Ntupling:
  - Convert into format suited for interactive analysis
  - Still too big for interactive analysis
- Skimming & Slimming:
  - Reduce number of events and information content



CHEP 2016: Proof of Principle

- Not changing the analysis workflow, optimizing the bookkeeping
    - Apache spark
    - Analyzer code in Scala
    - Input converted in Avro: https://github.com/diana-hep/rootconverter, stored on the HDFS



```
// Reference the whole dataset (not individual files)
val mcsample = avrordd("hdfs://path/to/mcsample/*.avro")

// First pass (and cache for later)
mcsample.persist()
val mc_sumOfWeights = mcsample.map(_.GenInfo.weight).sum

// Second pass on data in cluster's memory
val result = mcsample.filter(cuts).map(toNtuple(_, mc_sumOfWeights, mc_xsec))

// Save as ntuple
result.toDF().write.parquet("hdfs://path/to/mcsample_ntuple")
```

Input

Sum of Weights for Simulation

Main Event Selection

Output

Two loops over file entries, parallel jobs in Spark across cluster

Output ntuple is used for analysis e.g: plots, fits, tables

```
# Bring the ntuple in as a DataFrame
ntuple = spark.read.parquet("hdfs://path/to/mcsample_ntuple")

ntuple.select("mass").show()
...
```

Output contains information of:

- Object (e.g. Muon/Jet)
- Event (e.g. Luminosity) information

Physics plots!

Comparison

	Spark	ROOT
Analysis run without caching	9.4 sec	32.7 sec
Reading from local disk & Computation	*4.3 sec*	26.8 sec
Writing to local disk	5.1 sec	5.9 sec
Analysis run with caching	5.5 sec	
Reading from memory cache & Computation	*0.4 sec*	
Writing to local disk	5.1 sec	

- Running both the Spark workflow and ROOT workflow on a single CERN Lxplus node, using one core
- Input files on local disk: 1 GB ROOT file, 2 GB AVRO file Caveat: ROOT file is compressed, AVRO is not

Conclusion: comparing performances not easy; spark is not order of magnitudes slower

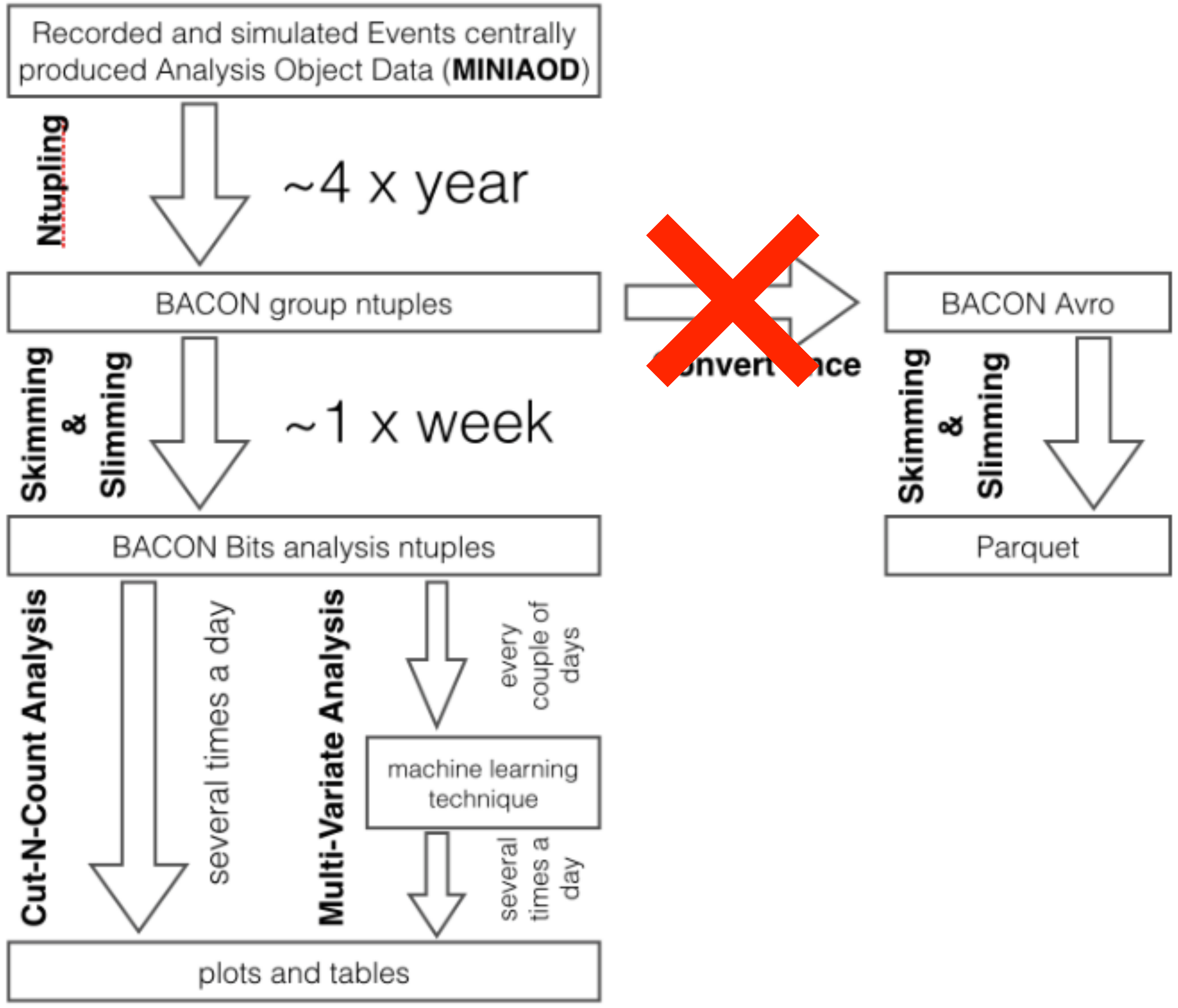
Next steps

- Thrust 1:
  - Use analysis-specific data formats that have all recipes applied and framework code re-run
  - Explore using Apache spark producing plots and tables
- Thrust 2:
  - Use official input
  - Demonstrate reduction capabilities producing group analysis ntuples
  - Goal: reduce 1 PB input to 1 TB output in 5 hours (CERN Openlab/Intel project)

Applying official recipes or re-running CMS framework code currently not being considered

Thrust 1

- https://github.com/diana-hep/spark-root
- Read ROOT files directly from Apache Spark
  - Connect ROOT to Apache Spark to be able to read ROOT TTrees, infer the schema and manipulate the data via Spark's DataFrames/Datasets/RDDs.



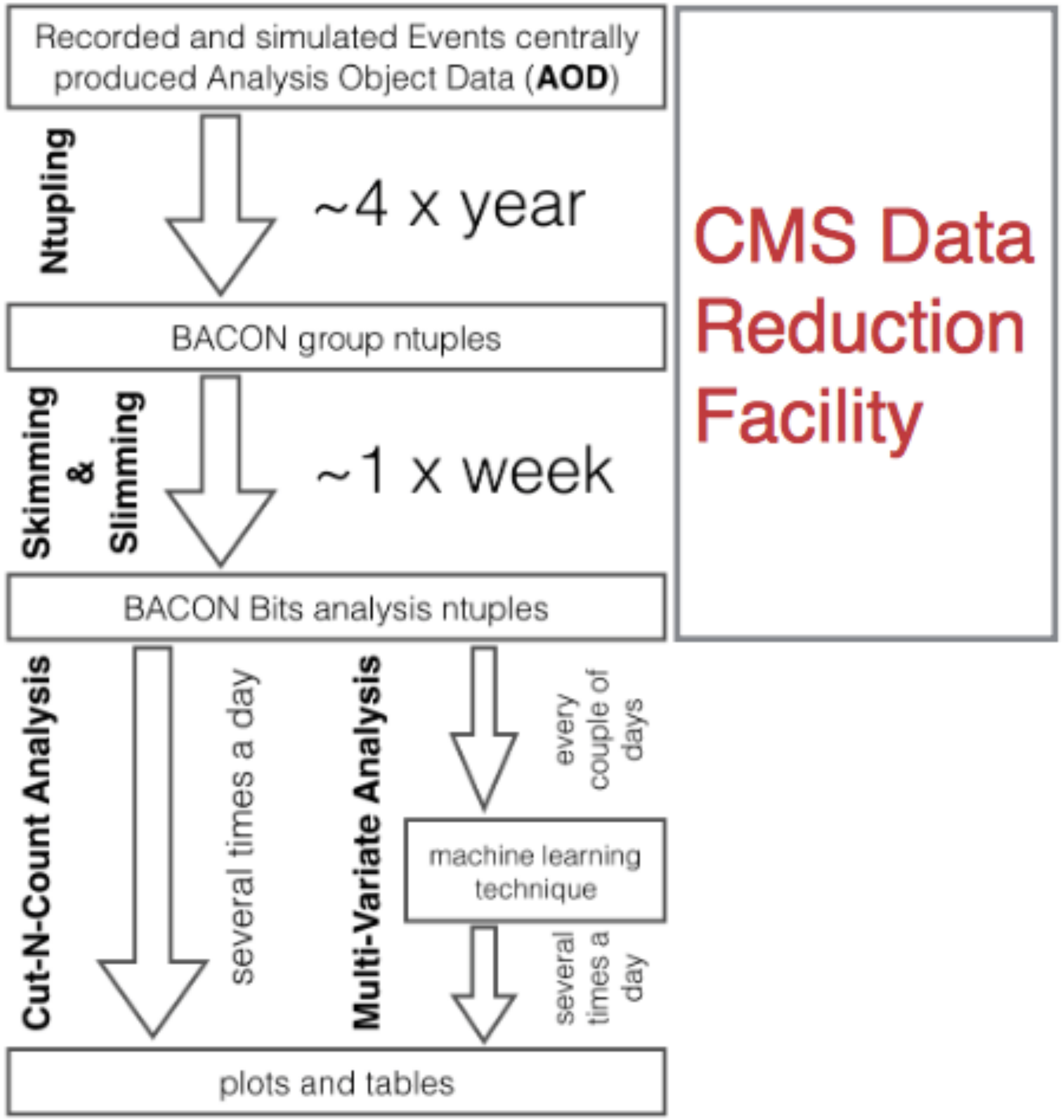
Scala code on Spark

more in Jim’s talk today

Thrust 2

Data Reduction Facility

- demonstrate the ability to reach at least a **1000 fold reduction** in selected data
- perform this task roughly **100 times faster** than it can currently be done
- Goal:
  - Process an input sample of 1PB within 5 hours
  - Export a selected sample that is at least 1000 times smaller
- Work on CERN Hadoop using Spark
  - Started with using CMS open data
  - Copied small amounts to HDFS (currently using 1.2 TB)
- Next step: enable Spark to read ROOT files through spark-root directly from HEP storage systems (EOS)



CMS Big Data Project

- Group created end of 2015
  - collaboration between FNAL, Diana-HEP, and CERN-IT
  - website: https://cms-big-data.github.io
- Rapidly expanding:
  - University of Padova and Bristol recently joined
- CERN Openlab enables partnership with industry:
  - CERN Openlab/Intel project called "CMS Data Reduction Facility"
  - Project includes CERN fellow supporting the development and testing of the reduction facility
  - Intel actively taking part in project
  - Sponsoring of CERN fellow included in the project

Backup